

APPLICATION FOR UNITED STATES LETTERS PATENT

Applicant:

Thomas E. Chefalas, et al.

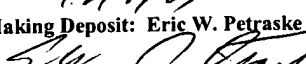
For:

**“Intelligent Caching of Working Directories in
Auxiliary Storage”**

Docket: YOR920030572US1

**INTERNATIONAL BUSINESS
MACHINES CORPORATION
ARMONK, NEW YORK 10504**

I HEREBY CERTIFY THAT THIS CORRESPONDENCE IS BEING DEPOSITED WITH THE
UNITED STATES POSTAL SERVICE AS EXPRESS MAIL IN AN ENVELOPE ADDRESSED
TO: U.S. PATENT AND TRADEMARK OFFICE, WASHINGTON, D.C. 20231 THE
APPLICANT AND/OR ATTORNEY REQUESTS THE DATE OF DEPOSIT AS THE FILING
DATE.

Express Mail No: ER0129171143US
Date of Deposit: 12/8/92
Name of Person Making Deposit: Eric W. Petraske
Signature: 

1 INTELLIGENT CACHING OF WORKING DIRECTORIES
2 IN AUXILIARY STORAGE

3 FIELD OF INVENTION

4 The present invention relates to a method and apparatus for the intelligent
5 caching of working directories in auxiliary or temporary storage, in
6 particular to battery-powered systems.

7 BACKGROUND OF THE INVENTION

8 Today's computer systems rely heavily on disk access to load programs,
9 store data and configuration information, and save files. Programs are so
10 large that they often can't fit into usable physical memory and often are
11 forced to run in logical memory partitions.

12 When a branch occurs in the program flow, the sequence of instructions
13 may call on an instruction that is located on the disk and not in the RAM or
14 cache memory. In that case, the program operation pauses while the disk
15 turns. When the correct portion of the disk is read, the relevant segment of
16 the program is read into RAM and execution resumes.

17 Much effort has gone into the development of cache memory and of

1 programming techniques to avoid excessive disk calls, but modern large
2 programs still make very many calls to instructions or data that are not
3 located in RAM or in the cache.

4 Programs such as Internet Explorer rely on the hard disk to cache hundreds,
5 even thousands of files such as GIFs, JPGs, HTML pages, etc. This causes a
6 great deal of disk activity, which not only is slow but uses a
7 correspondingly large amount of battery power. Some programs create and
8 use a large number of temporary files. All of this causes a great deal of disk
9 activity, causing the hard drive to keep running or continuously spin up and
10 down, wasting power and delaying operation of the program until the
11 instructions or data are fetched from the disk.

12 In addition to technical reasons for the great amount of disk usage, there are
13 also marketing considerations. Quite often, retailers will advertise computer
14 systems with a minimal amount of RAM, though having very large capacity
15 disk drives. The lack of RAM aggravates the problem of excessive disk
16 usage because there can be less data and instructions stored in a smaller
17 amount of RAM.

18 If the only computer systems affected were desk-top systems, the main
19 drawback to these various trends would be slower performance of the
20 programs being run. The trend to greater use of portable computer systems
21 means that excessive disk calls translate very quickly to short battery life.

22 It is not realistic to expect that programmers will spend additional effort to

1 shrink the size of the programs. It is a truism in programming that it takes
2 longer to code a compact program than a bloated one. With present
3 competitive pressures to introduce features and to bring out frequent
4 upgrades, it is not realistic to expect that program vendors will shrink the
5 size and storage demands of their products in the near future.

6 Methods of reducing disk access are known, such as cache memory, which
7 employs relatively high power consumption integrated circuits. Other
8 techniques could be used that would consume less power than conventional
9 cache memory.

10 As mentioned above, the computer vendors ordinarily emphasize disk
11 capacity in their advertising and not memory size.

12 Operating systems could take advantage of technical features of relatively
13 low power alternatives to disk drives, such as flash memory or other solid
14 state memory.

15 There is a chicken and egg problem in that the programmers of application
16 programs will not spend the time to offer such features until a large fraction
17 of systems have the memory devices to take advantage of the possibility of
18 saving battery power and execution time and hardware vendors will not add
19 to their base cost in a highly competitive market unless that extra cost can
20 produce a benefit in battery life or program response that the consumer is
21 willing to pay for.

1 Thus, there is a potential benefit to the consumer from a third-party offering
2 of a combination of non-rotating hardware and associated software that will
3 improve the performance of application programs without waiting for the
4 application programmers and/or the operating system programmers to offer
5 such features.

6 SUMMARY OF THE INVENTION

7 The invention relates to a combination of non-rotating storage and
8 associated software to temporarily transfer storage functions from a hard
9 disk to the non-rotating storage.

10 This invention provides for the "on the fly" adaptation and transfer of an
11 application's disk requirements to be instead sent to a flash storage,
12 non-spinning memory device, solid state memory, or similar device while
13 the hard drive is powered down. This not only makes operations faster, but
14 results in a large savings in battery power.

15 BRIEF DESCRIPTION OF THE DRAWINGS

16 FIG. 1 illustrates a block diagram of the present invention.

17 DETAILED DESCRIPTION

1 Figure 1 illustrates in block diagram form a battery-powered computer
2 system 101 having an auxiliary agent program 102 that will be described
3 below. The computer labeled system 101 schematically includes the
4 computer CPU, disk storage, RAM, and conventional caches. Hard disk
5 drive 103 and temporary storage 105 are also shown.

6 The agent program 102 communicates with a database 103 that may be
7 conveniently located in RAM or in temporary storage 105. The function of
8 data base 110 is to keep track of the location of data and program segments
9 that would ordinarily be placed on the hard drive.

10 In operation, large programs move segments of the program from the long-
11 term storage on the hard drive into RAM, to improve access time. There is
12 not enough room in RAM to hold many popular programs, plus the
13 operating system and possibly other programs. Thus, the typical large
14 application program (or, equivalently a portion of the operating system) will
15 contain program language that recognizes when the application program
16 needs data or instructions that are not in RAM or in the CPU cache and
17 fetch the required information from the disk, while the program pauses.

18 The database 110 of the present invention is activated when the computer
19 system 101 is started and immediately begins monitoring programs and the
20 files associated with the programs.

21 When a program is started that creates or edits files, the associations
22 between that program and the file or files it creates or edits is stored in an

1 associative database as we have described in copending patent application
2 Docket YOR8-2001-0031, incorporated herein by reference in its entirety.
3 Such a database is required to maintain the association of an application to
4 its files, without regard to the file type extension or naming method.
5 Normally, an operating system such as Microsoft Windows keeps track of
6 files as types by the value of the file extension, which is the group of
7 characters following the rightmost "." in the file name.

8 The database 110 according to the present invention maintains the
9 association of files and programs in a separate system from the operating
10 system.

11 The database provides an important link from the application through the
12 database to the data files. If a temporary file or folder is created, that folder
13 or file is associated with that application. When the application is closed,
14 the contents of those files and folders in the temporary storage are written to
15 the main hard disk storage. When the application is re-run, those temporary
16 folders and files are created on behalf of the application in the temporary
17 storage.

18 Subsequent access to the folders or files is made directly to the temporary
19 storage. The contents of the temporary store are then written out to the hard
20 disk when the program exits.

21 With this approach, the benefits of fast, low power storage may be realized
22 without altering the operating system.

1 When the computer system 101 is started, the invention database agent 102
2 begins monitoring the computer system 101. When a program (e. g. an
3 application program) is launched on computer system 102, the database
4 agent 102 of the present invention begins monitoring the program and keeps
5 track of its storage requests.

6 If the program has been previously started and has created files or folders,
7 those files or folders are read from the hard disk database 103 and placed
8 into the temporary storage device 105.

9 If the program has been written such that it creates (temporary) files or
10 folders each time it is started, those files or folders are also placed into the
11 temporary storage device 105.

12 The agent also copies program segments that are most frequently used (and
13 are not in RAM or in the operating system's cache memory) from the hard
14 drive to the temporary storage, using stored information kept by the
15 program, if such information is available, or alternatively using its own
16 records on the most-often used segments.

17 The invention then attempts to power down the hard disk drive in the
18 computing system 101. If no other applications require the hard drive to be
19 running, the drive may be powered down so that it is no longer rotating and
20 using battery power.

21 The agent of the current invention 102 monitors the file system operations

1 by installing an intercept hook into the operating system file functions.
2 When it receives requests from the file system, the agent routes these
3 requests to the temporary memory 105. It returns the status to the operating
4 system, and the operating system thinks it has performed the operation to
5 the actual hard disk drive. When the program is finished or ends, the hard
6 disk drive is started, if necessary, and the contents of the temporary drive
7 are reflected back to the actual hard drive. The temporary storage 105 is also
8 initialized for reuse.

9 An important advantage of the present invention is that it does not require
10 re-writing the disk handling portion of the operating system. Thus, the use
11 of the invention is not contingent on a business decision by the operating
12 system vendor.

13 If no space is available on the temporary storage 105 or in the course of
14 operation the temporary storage device 105 becomes full, the hard disk
15 drive is started, if necessary, and the contents of the temporary storage
16 device 105 are written to the hard disk drive. Thus power savings are
17 accomplished as the size of temporary storage permits, but the lack of
18 temporary storage space will still allow applications to run, but without
19 benefit of the power saving measures.

20 As a simple example of just a few uses of the invention, consider a word
21 processor. When the program is first called during a work session, the
22 computer operating system will load a portion of the program into RAM and
23 into the computer's cache.

1 When the user starts work on a new document, the program stores the new
2 text in RAM and, from time to time, backs up the work done to date on the
3 temporary storage 105 instead of the hard drive. When the user inserts a
4 mathematical symbol into a document or switches to a different font, the
5 application program will attempt to fetch the required data. If the data are
6 not in RAM, the operating system will try the cache and then call to the
7 disk.

8 The agent will intercept the disk call and check if the relevant data have
9 been loaded into the temporary storage. If the relevant symbol library
10 and/or font definition have been used in the past fairly often, the agent will
11 have loaded them into temporary storage 105 during the program setup for
12 this work session and the program segment currently running in RAM will
13 be automatically directed to temporary storage 105 instead of powering up
14 disk 103, thus saving both the battery power and time required to bring the
15 disk up to operating speed.

16 If the relevant data have not been pre-loaded, the agent will not interfere
17 with the call by the program (operating system) to the disk, but will store the
18 data retrieved in temporary storage 105. Conventional algorithms used in
19 cache memory may be applied here to decide how much data in the vicinity
20 of the data that is specifically required will be stored in temporary storage.

21 In the particular illustrated example, a user who calls up the Greek symbol
22 "nu" may well also call up "pi" and "lambda". Associative data may be kept
23 in the database and stored between work sessions to build up a picture of

1 this particular user's ordinary practices. Third party sources of information
2 as to what features are most commonly used, such as the program vendor or
3 the local IT department, may also be available.

4 The agent has been illustrated as being separate from the operating system.
5 Those skilled in the art will appreciate that an agent according to the
6 invention could also be incorporated into the operating system.

7 The instructions for the agent may be located on non-volatile storage, e.g. a
8 disk, and copied into RAM; or the instructions could be stored in non-
9 volatile solid state memory in the same hardware as the storage 105.

10 For convenience in the following claims, the term "data" will be used
11 generally to cover both items such as the text in a word processor or the
12 numbers in a spreadsheet and also the instructions that are part of an
13 application program: i.e. the agent may store program segments and/or text
14 or numbers in the temporary storage.

15 It should be noted that this invention is not a RAM disk, though it may
16 make use of a RAM disk for the non-rotating storage. The technology for
17 RAM disks, in which a portion of RAM is set up to emulate the structure of
18 a disk, has been around for many years and is well understood.

19 The non-rotating storage 105 is preferably the storage in the system that
20 consumes the least power. Alternatives to RAM are a network storage
21 device, an SAN (storage area network) device, organic memory, EEPROM,

1 flash memory and others.

2 Since the main function of the invention is saving power, the initial cost of
3 the components in the storage 105 will weigh less in the mind of the user
4 than the benefits of longer battery life.

5 Optionally, the agent may reformat the intercepted data being sent by the
6 operating system to the hard drive. For example, if the operating system
7 converts a text file to segments reflecting the physical structure of the disk,
8 the agent may either store the data on storage 105 in the same format, or
9 convert it to any other format that may be convenient. If the operating
10 system sends out data in the same format that the application program uses
11 and electronics on the disk drive formats the data as required by the disk,
12 the agent may simply keep the format.

13 While the invention has been described in terms of a single preferred
14 embodiment, those skilled in the art will recognize that the invention can be
15 practiced in various versions within the spirit and scope of the following
16 claims.